Pizza Sales Analysis

An End-to-End Azure Data Engineering Project

Objective:

To analyze pizza sales data by creating a modern, cloud-based data engineering pipeline, leveraging Azure services and Power BI for actionable insights**.**

Key Features:

1. **Data Migration**: Seamless transfer of pizza sales data from SQL Server to Azure Blob Storage using Azure Data Factory**.**
2. **Data Processing:** Advanced transformations and aggregations on Azure Databricks using PySpark and Spark SQL.
3. **Data Visualization:** Insightful dashboards in Power BI showcasing KPIs such as total sales, pizza popularity, and revenue trends.

Technologies Used:

- **SQL Server:** Source database**.**
- **Azure Data Factory:** Data pipeline orchestration.
- **Azure Blob Storage:** Scalable cloud storage.
- **Azure Databricks:** Data transformation and aggregation.
- **Power BI:** Visualization and reporting.

Prepared By:

Alkesh Lajurkar
LinkedIn: linkedin.com/in/alkeshlajurkar
GitHub: github.com/alkeshlajurkar
Email: alkeshlajurkar@gmail.com
Contact: +91 8390956095

---

1. Introduction

This project demonstrates how to design and implement an end-to-end data engineering pipeline. The pipeline includes migrating data from an on-premises SQL Server to Azure, processing it with Azure Databricks, and visualizing insights in Power BI. This comprehensive guide walks through the steps to implement the solution while ensuring scalability, automation, and actionable insights.

Skills Demonstrated

- SQL Server
- Azure Data Factory
- Azure Databricks (PySpark & Spark SQL)
- Power BI
- End-to-End Data Engineering Pipeline

---

2. Business Requirement Overview

Objective

The goal is to build a pipeline that efficiently migrates and processes data from on-prem SQL Server to Azure for business reporting. The final output is a Power BI dashboard presenting KPIs and trends derived from the data.

Tasks Overview

1. Import raw sales data from CSV to SQL Server.
2. Migrate data from SQL Server to Azure Blob Storage using Azure Data Factory.
3. Process and transform the data in Azure Databricks.
4. Visualize the processed data in Power BI.

Tools Used

- **SQL Server:** On-prem database for raw data.
- **Azure Storage Account:** Cloud storage to hold migrated data.
- **Azure Data Factory (ADF):** Tool for orchestrating data movement.
- **Azure Databricks:** For data processing and KPI generation.
- **Power BI:** Visualization tool for creating dashboards.

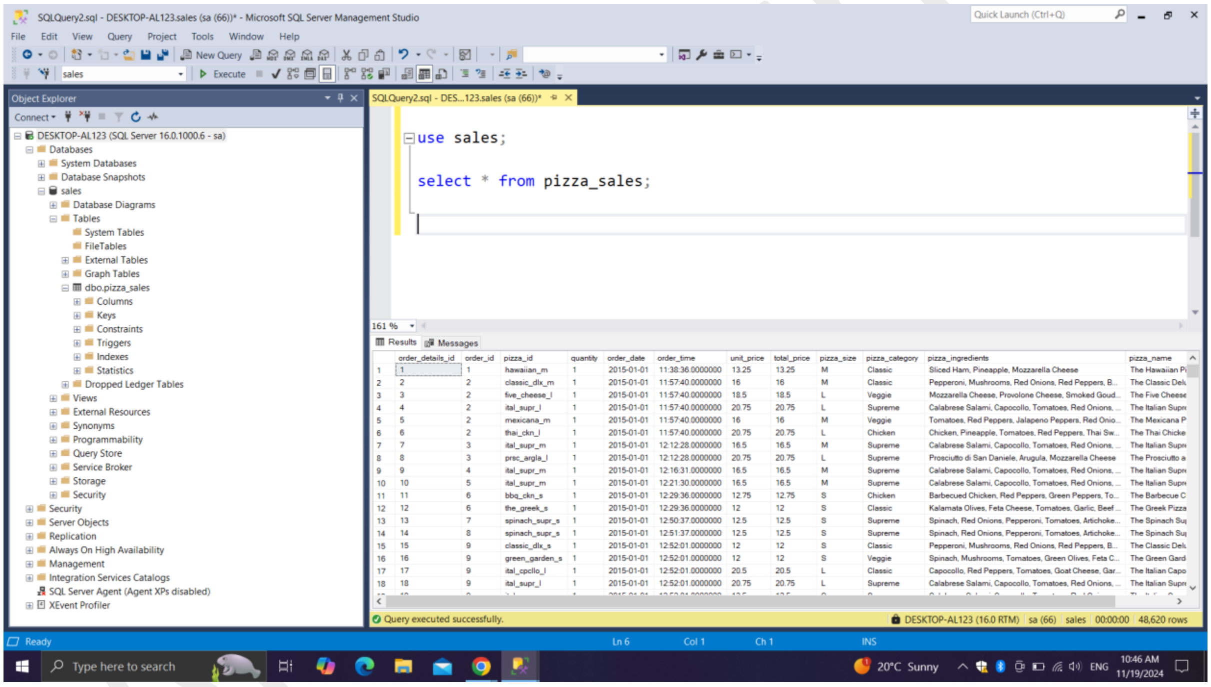
---

3. Step-by-Step Implementation

Step 1: Import CSV Data to SQL Server

Objective: Load raw sales data into a SQL Server database for processing.

1. **Create the Database:** Use SQL Server Management Studio (SSMS) to create a database.
2. **Create the Table:** Define a table named PizzaSales with columns to store sales data.
3. **Import Data from CSV:**
    - Use SSMS's *Import Flat File Wizard* to upload data from a CSV file.
    - Map columns correctly and execute the process.
4. **Verify Data:** Run a query to ensure the data was imported successfully.



```sql
use sales;

select * from pizza_sales;
```

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pizza_name
1	1	1	hawaiian_m	1	2015-01-01	11:38:36.0000000	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pi
2	2	2	classic_dlx_m	1	2015-01-01	11:57:40.0000000	16	16	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, B...	The Classic Delu
3	3	2	five_cheese_l	1	2015-01-01	11:57:40.0000000	18.5	18.5	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Goud...	The Five Cheese
4	4	2	ital_supr_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The Italian Supre
5	5	2	mexicana_m	1	2015-01-01	11:57:40.0000000	16	16	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red Onio...	The Mexicana P
6	6	2	thai_ckn_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sw...	The Thai Chicke
7	7	3	ital_supr_m	1	2015-01-01	12:12:28.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The Italian Supre
8	8	3	prsc_argla_l	1	2015-01-01	12:12:28.0000000	20.75	20.75	L	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella Cheese	The Prosciutto a
9	9	4	ital_supr_m	1	2015-01-01	12:16:31.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The Italian Supre
10	10	5	ital_supr_m	1	2015-01-01	12:21:30.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The Italian Supre
11	11	6	bbq_ckn_s	1	2015-01-01	12:29:36.0000000	12.75	12.75	S	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, To...	The Barbecue C
12	12	6	the_greek_s	1	2015-01-01	12:29:36.0000000	12	12	S	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garlic, Beef ...	The Greek Pizza
13	13	7	spinach_supr_s	1	2015-01-01	12:50:37.0000000	12.5	12.5	S	Supreme	Spinach, Red Onions, Pepperoni, Tomatoes, Artichoke...	The Spinach Su
14	14	8	spinach_supr_s	1	2015-01-01	12:51:37.0000000	12.5	12.5	S	Supreme	Spinach, Red Onions, Pepperoni, Tomatoes, Artichoke...	The Spinach Su
15	15	9	classic_dlx_s	1	2015-01-01	12:52:01.0000000	12	12	S	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, B...	The Classic Delu
16	16	9	green_garden_s	1	2015-01-01	12:52:01.0000000	12	12	S	Veggie	Spinach, Mushrooms, Tomatoes, Green Olives, Feta C...	The Green Gard
17	17	9	ital_cpcllo_l	1	2015-01-01	12:52:01.0000000	20.5	20.5	L	Classic	Capocollo, Red Peppers, Tomatoes, Goat Cheese, Gar...	The Italian Capo
18	18	9	ital_supr_l	1	2015-01-01	12:52:01.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The Italian Supre

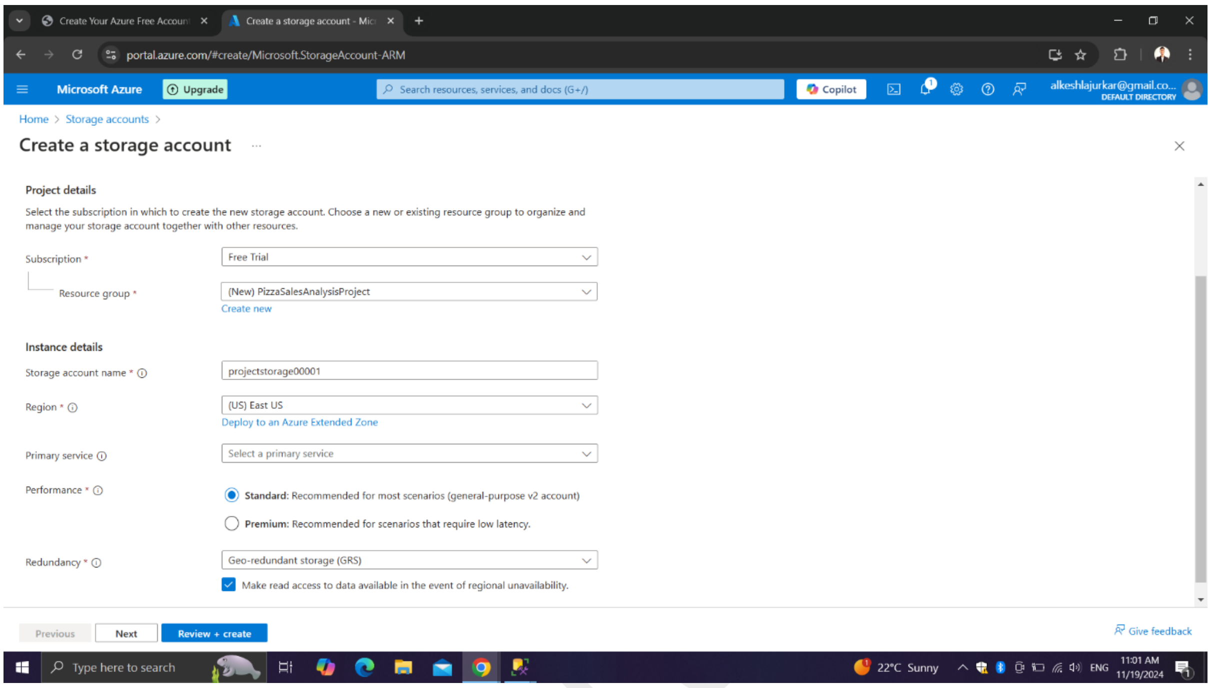
```
Query executed successfully.    DESKTOP-AL123 (16.0 RTM) | sa (66) | sales | 00:00:00 | 48,620 rows
Ln 6    Col 1    Ch 1    INS
```

Step 2: Set Up Azure Resources

Objective: Prepare Azure services for cloud-based data storage and processing.

1. **Create a Storage Account:**
    - Go to the Azure portal.
    - Create a storage account under your resource group.
    - Configure settings for secure data storage.

2. **Create a Blob Container:**
    - Inside the storage account, create a new container named salesdata.
    - This will store the raw files after migration.



portal.azure.com/#create/Microsoft.StorageAccount-ARM

Home > Storage accounts >

Create a storage account

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *	Free Trial
Resource group *	(New) PizzaSalesAnalysisProject

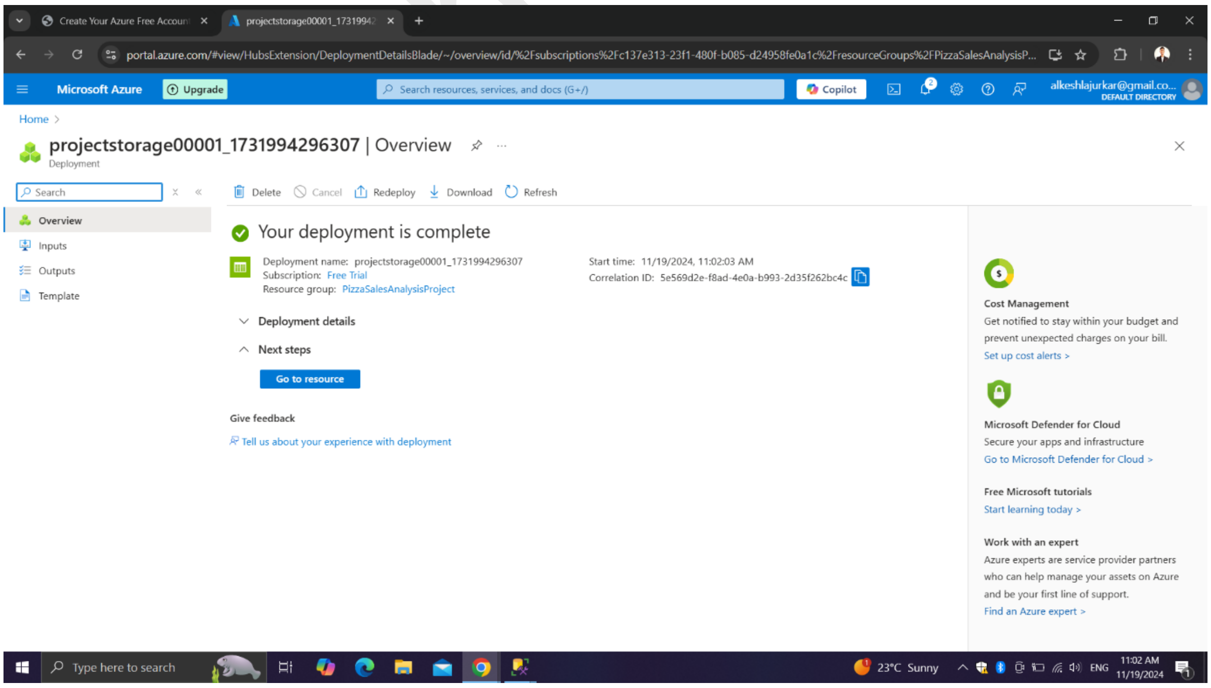
Create new

Instance details

Storage account name *	projectstorage00001
Region *	(US) East US
Primary service	Select a primary service
Performance *	[✓ Standard: Recommended for most scenarios (general-purpose v2 account)] Premium: Recommended for scenarios that require low latency.
Redundancy *	Geo-redundant storage (GRS)

Deploy to an Azure Extended Zone

[✓] Make read access to data available in the event of regional unavailability.



portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Fc137e313-23f1-480f-b085-d24958fe0a1c%2FresourceGroups%2FPizzaSalesAnalysisP...

Home >

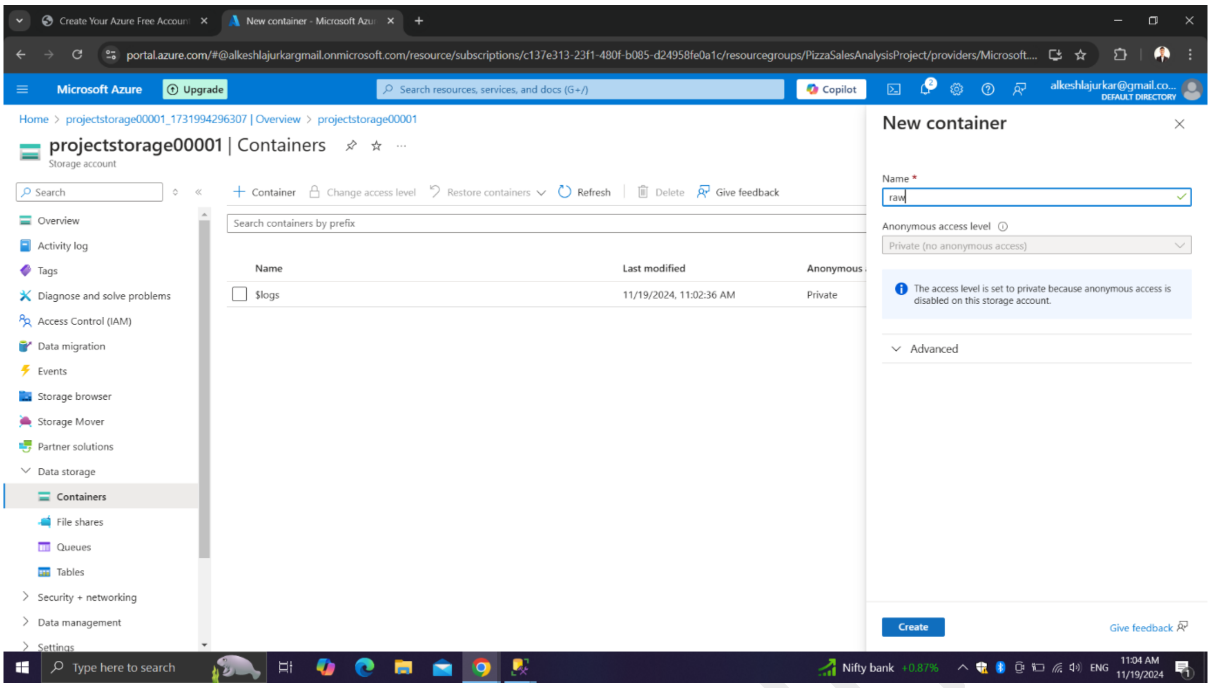
projectstorage00001_1731994296307 | Overview

Deployment

Your deployment is complete

Deployment name: projectstorage00001_1731994296307
Subscription: Free Trial
Resource group: PizzaSalesAnalysisProject

Start time: 11/19/2024, 11:02:03 AM
Correlation ID: 5e569d2e-f8ad-4e0a-b993-2d35f262bc4c


portal.azure.com/#@alkeshlajurkargmail.onmicrosoft.com/resource/subscriptions/c137e313-23f1-480f-b085-d24958fe0a1c/resourcegroups/PizzaSalesAnalysisProject/providers/Microsoft....

Home > projectstorage00001_1731994296307 | Overview > projectstorage00001

projectstorage00001 | Containers
Storage account

Name	Last modified	Anonymous
$logs	11/19/2024, 11:02:36 AM	Private

New container

Name *: raw

Anonymous access level: Private (no anonymous access)

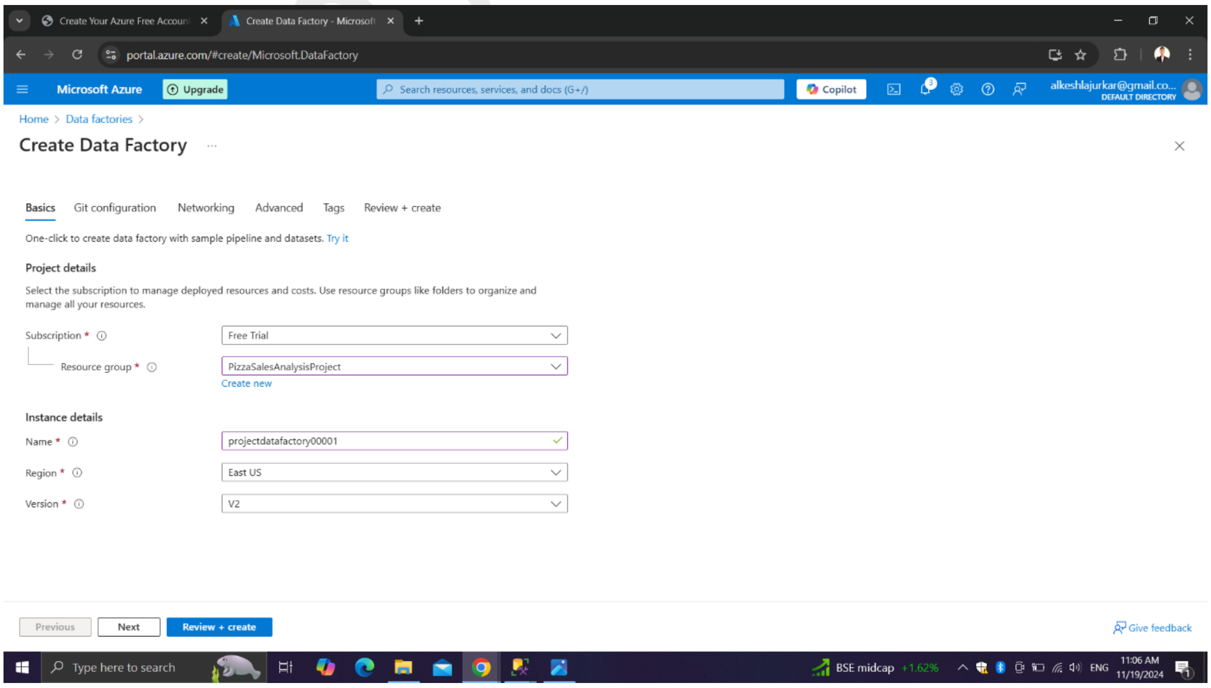
The access level is set to private because anonymous access is disabled on this storage account.

Advanced

Step 3: Migrate Data Using Azure Data Factory

Objective: Automate the transfer of data from SQL Server to Azure Blob Storage.

1. **Create an Azure Data Factory Instance:**
   - In the Azure portal, create a new Azure Data Factory instance.
   - Name it appropriately and configure the region.


portal.azure.com/#create/Microsoft.DataFactory

Home > Data factories >

Create Data Factory

Basics | Git configuration | Networking | Advanced | Tags | Review + create

One-click to create data factory with sample pipeline and datasets. Try it

Project details

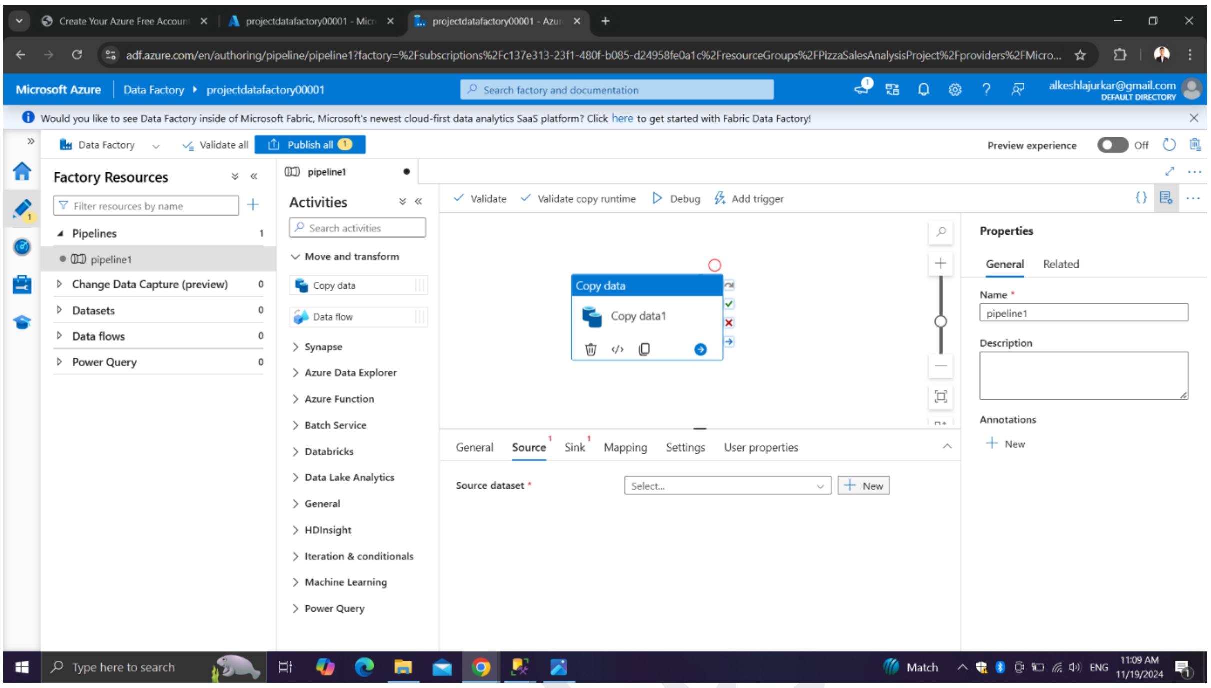
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Free Trial
Resource group *	PizzaSalesAnalysisProject

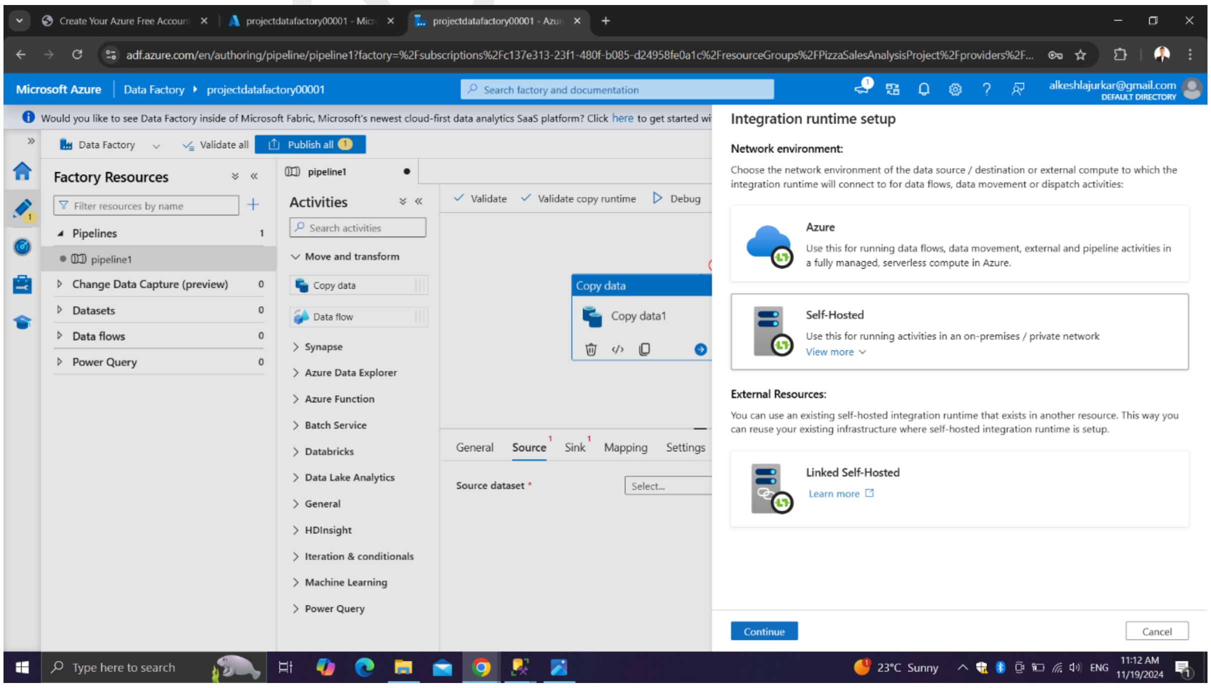
Instance details

Name *	projectdatafactory00001
Region *	East US
Version *	V2

2. **Build the Pipeline:**
   - Use the *Copy Data Activity* to create a pipeline that transfers data from SQL Server to Blob Storage.
   - Map source (SQL Server) and destination (Blob Storage).



3. **Configure Integration Runtime:**
   - Set up a self-hosted integration runtime to securely connect on-prem SQL Server to Azure.



Integration runtime setup

Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:

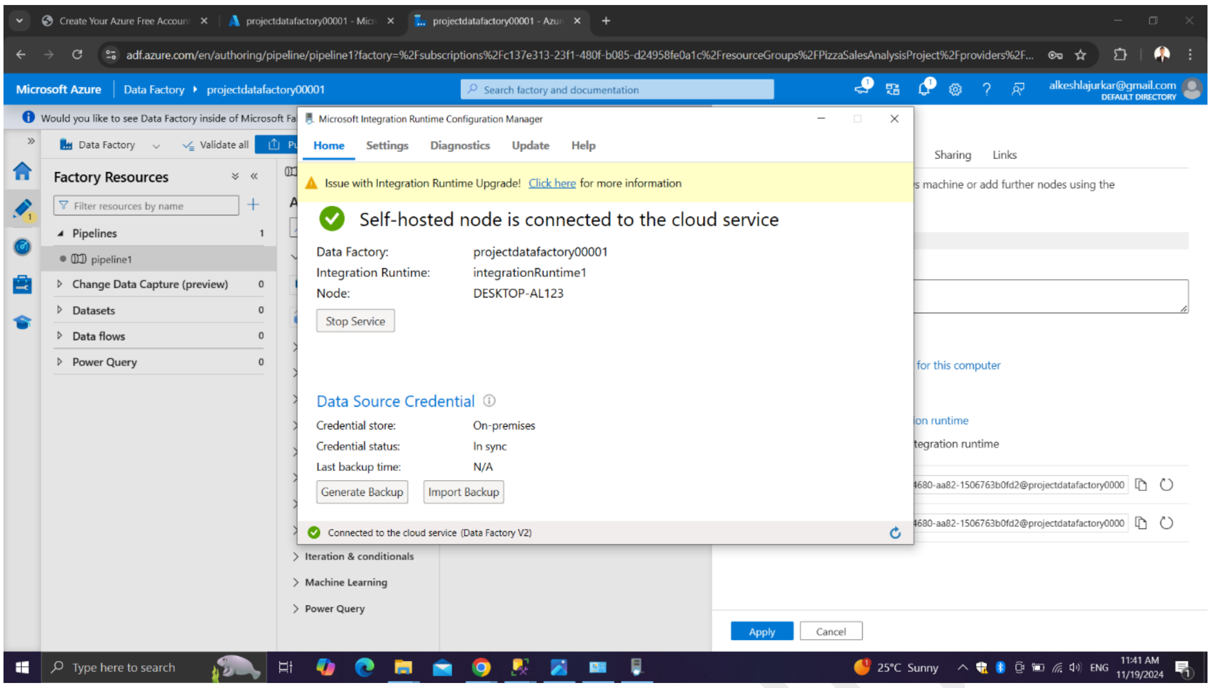
Azure
Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.

Self-Hosted
Use this for running activities in an on-premises / private network
View more

External Resources:

You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.

Linked Self-Hosted
Learn more


Microsoft Integration Runtime Configuration Manager

Issue with Integration Runtime Upgrade! Click here for more information

Self-hosted node is connected to the cloud service

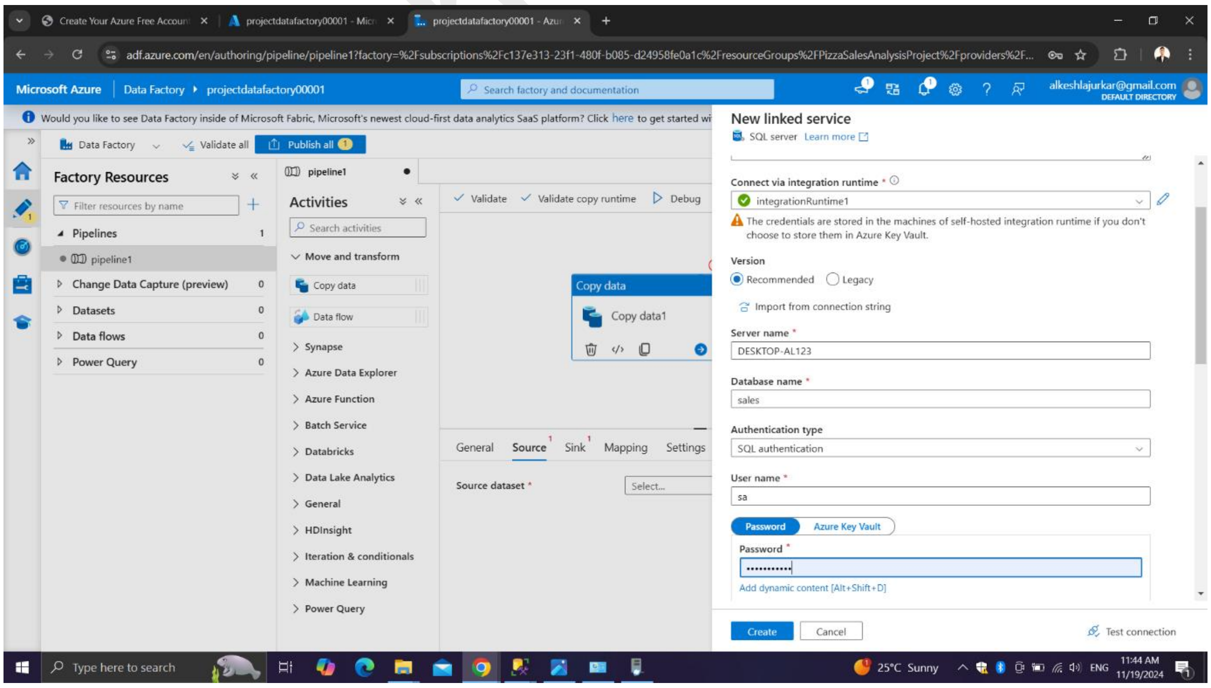
Data Factory:	projectdatafactory00001
Integration Runtime:	integrationRuntime1
Node:	DESKTOP-AL123

Data Source Credential

Credential store:	On-premises
Credential status:	In sync
Last backup time:	N/A

Connected to the cloud service (Data Factory V2)

4. **Set Up Linked Services:**
   - **SQL Server Linked Service:** Connect to your on-prem SQL Server.
   - **Blob Storage Linked Service:** Connect to your Azure Blob Storage.


New linked service

SQL server Learn more

Connect via integration runtime *: integrationRuntime1

The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Version: [✓ Recommended] Legacy

Import from connection string

Server name *: DESKTOP-AL123

Database name *: sales

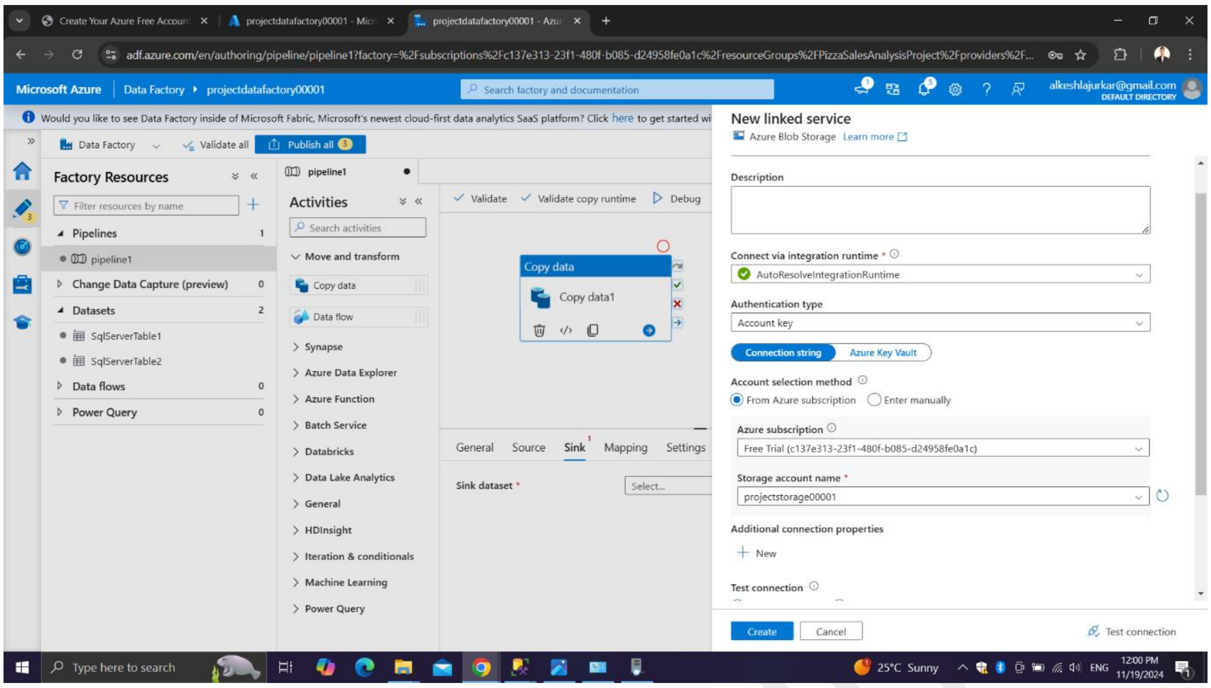
Authentication type: SQL authentication

User name *: sa

[✓ Password] Azure Key Vault

Password *: •••••••••••

Add dynamic content [Alt+Shift+D]


New linked service

Azure Blob Storage

Description

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication type: Account key

[✓ Connection string] Azure Key Vault

Account selection method: [✓ From Azure subscription] Enter manually

Azure subscription: Free Trial (c137e313-23f1-480f-b085-d24958fe0a1c)

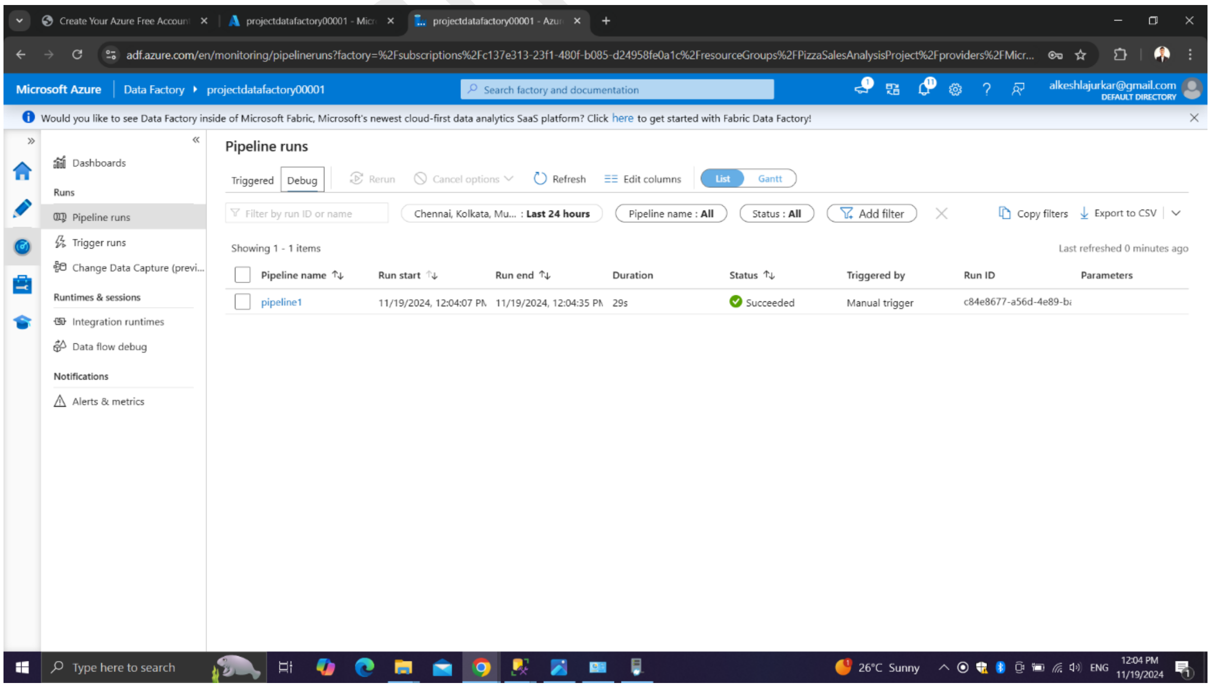
Storage account name *: projectstorage00001

Additional connection properties

Test connection

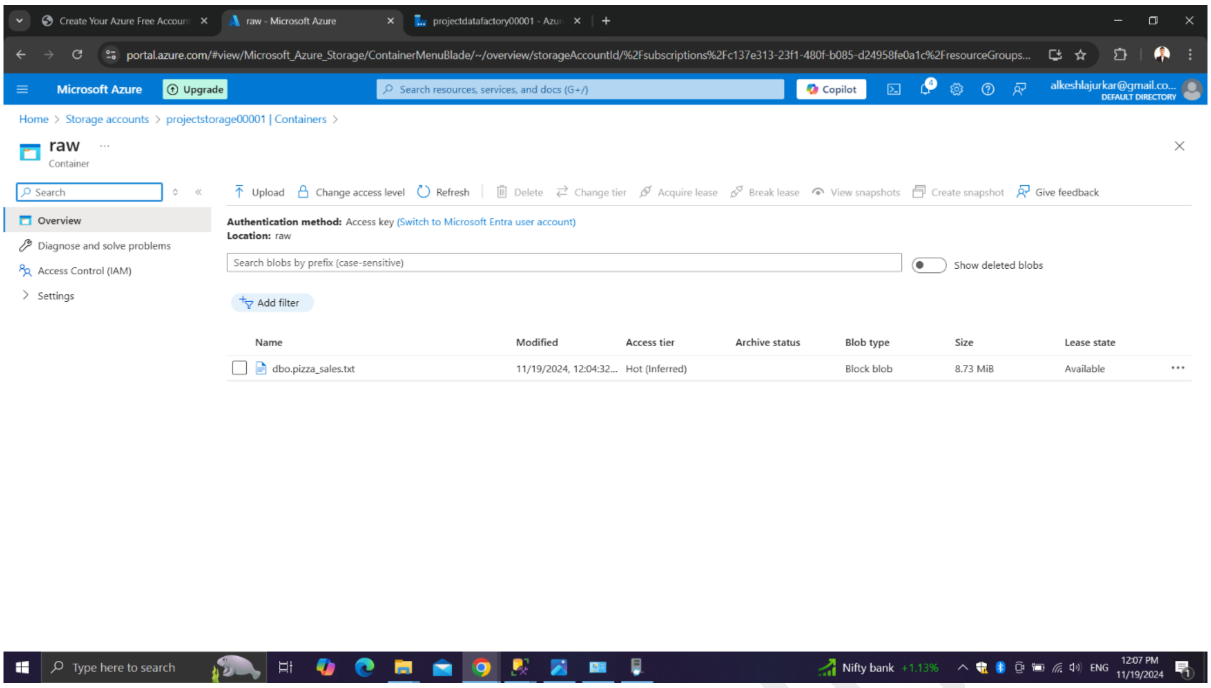
5. **Run and Verify the Pipeline:**

   - Execute the pipeline and verify that the raw file has been successfully transferred to the salesdata container in Blob Storage.


Pipeline runs

Showing 1 - 1 items

Pipeline name	Run start	Run end	Duration	Status	Triggered by	Run ID	Parameters
pipeline1	11/19/2024, 12:04:07 PM	11/19/2024, 12:04:35 PM	29s	Succeeded	Manual trigger	c84e8677-a56d-4e89-b	


Home > Storage accounts > projectstorage00001 | Containers >

raw
Container

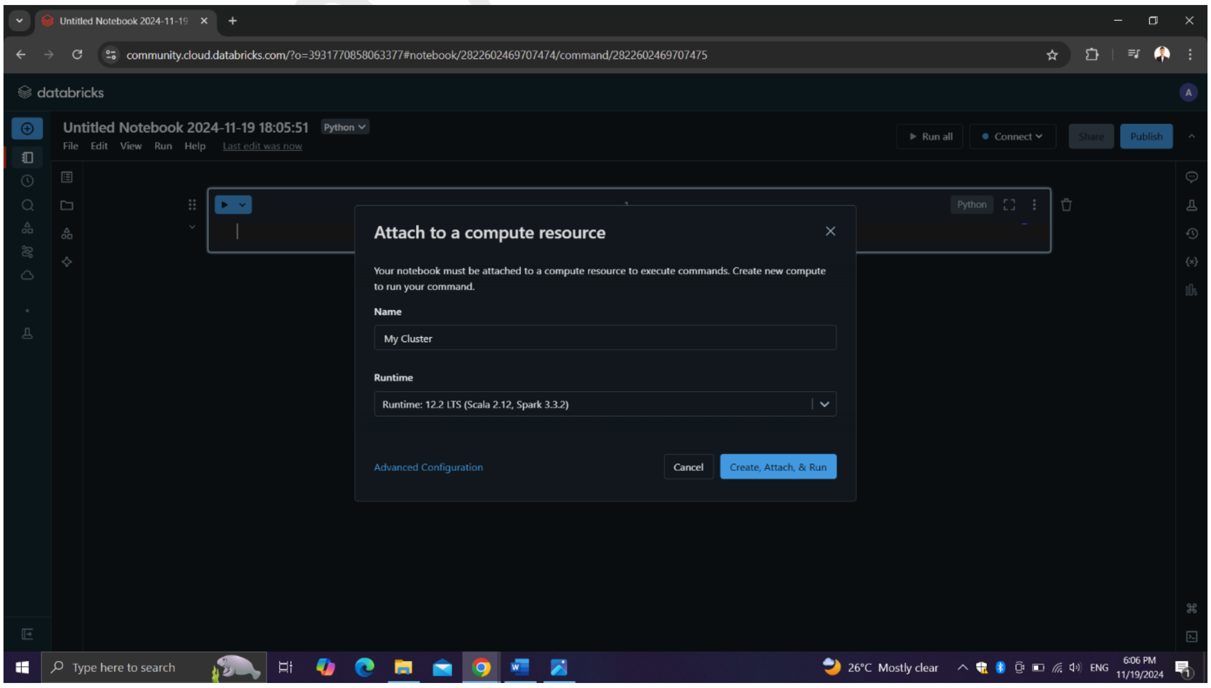
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: raw

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
dbo.pizza_sales.txt	11/19/2024, 12:04:32...	Hot (Inferred)		Block blob	8.73 MiB	Available

Step 4: Data Processing in Azure Databricks

Objective: Process raw data and generate KPIs using Spark-based workflows.

1. **Set Up Azure Databricks Workspace:**
   - In the Azure portal, create a new Databricks workspace.
   - Launch the workspace and create a new cluster.


community.cloud.databricks.com/?o=3931770858063377#notebook/2822602469707474/command/2822602469707475

Untitled Notebook 2024-11-19 18:05:51

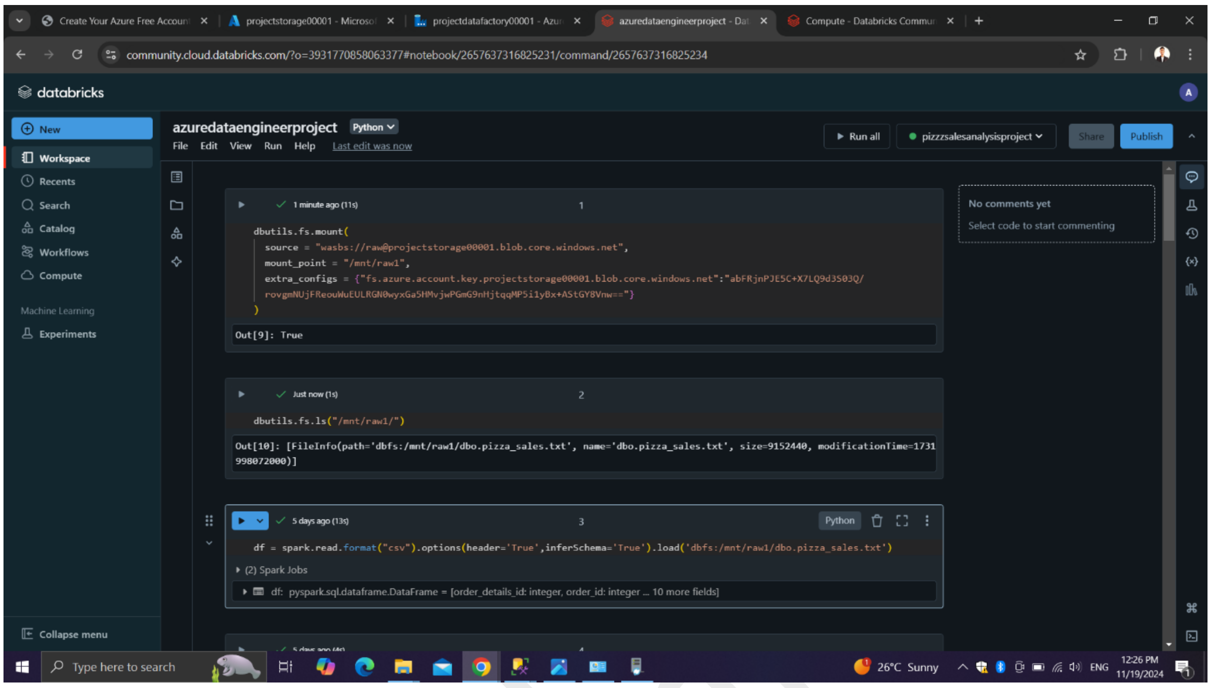
Attach to a compute resource

Your notebook must be attached to a compute resource to execute commands. Create new compute to run your command.

Name: My Cluster

Runtime: Runtime: 12.2 LTS (Scala 2.12, Spark 3.3.2)

2. **Mount Blob Storage in Databricks:**
   - Use the following PySpark code to mount the salesdata container in Databricks.



```
dbutils.fs.mount(
  source = "wasbs://raw@projectstorage00001.blob.core.windows.net",
  mount_point = "/mnt/raw1",
  extra_configs = {"fs.azure.account.key.projectstorage00001.blob.core.windows.net":"abFRjnPJE5C+X7LQ9d3S03Q/
  rovgmNUjFReouWuEULRGN0wyxGa5HMvjwPGmG9nHjtqqMP5i1yBx+AStGY8Vnw=="}
)
```

```
Out[9]: True
```

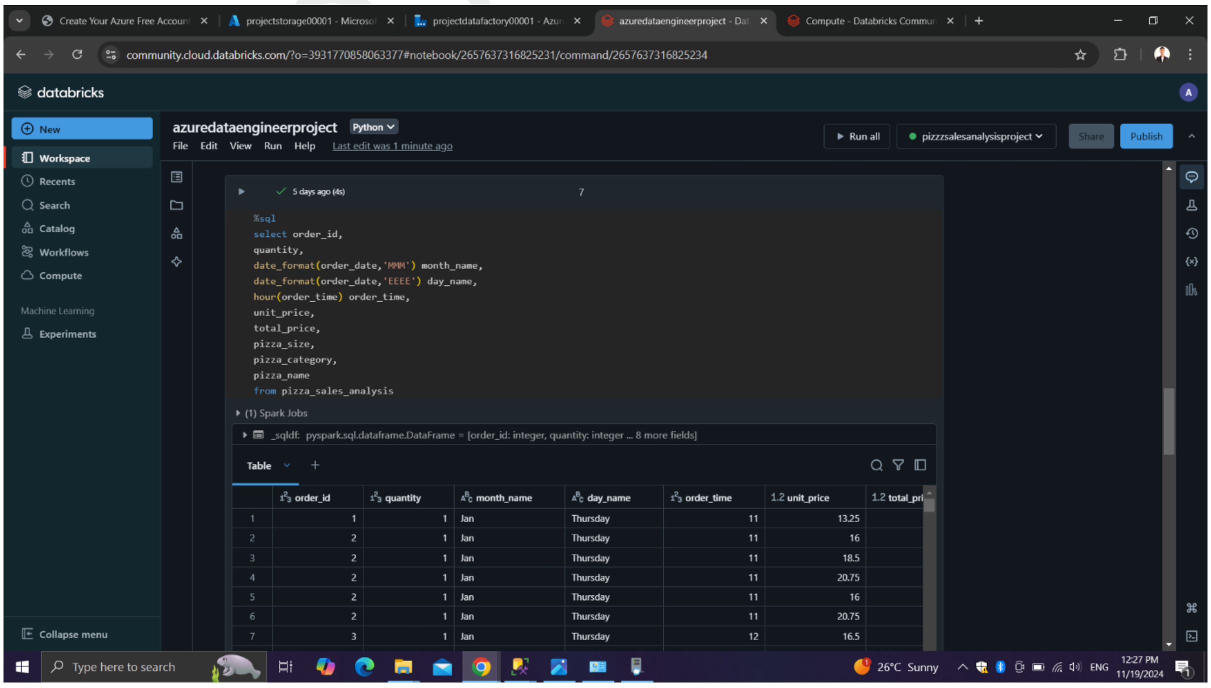
```
dbutils.fs.ls("/mnt/raw1/")
```

```
Out[10]: [FileInfo(path='dbfs:/mnt/raw1/dbo.pizza_sales.txt', name='dbo.pizza_sales.txt', size=9152440, modificationTime=1731
998072000)]
```

```
df = spark.read.format("csv").options(header='True',inferSchema='True').load('dbfs:/mnt/raw1/dbo.pizza_sales.txt')
```

```
(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [order_details_id: integer, order_id: integer ... 10 more fields]
```

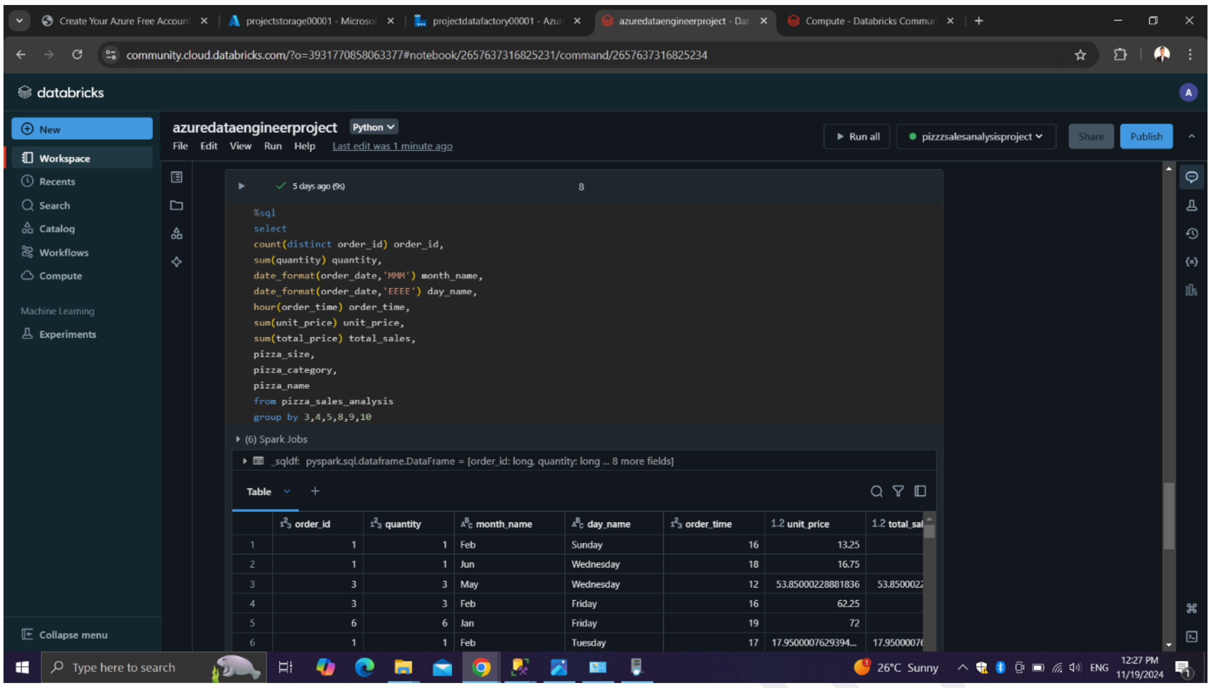
3. **Read Data into a PySpark DataFrame:**
4. **Perform Data Transformations:**
   - Calculate KPIs such as total sales and total quantity.



```
%sql
select order_id,
quantity,
date_format(order_date,'MMM') month_name,
date_format(order_date,'EEEE') day_name,
hour(order_time) order_time,
unit_price,
total_price,
pizza_size,
pizza_category,
pizza_name
from pizza_sales_analysis
```

```
(1) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [order_id: integer, quantity: integer ... 8 more fields]
```

	order_id	quantity	month_name	day_name	order_time	unit_price	total_pri
1	1	1	Jan	Thursday	11	13.25	
2	2	1	Jan	Thursday	11	16	
3	2	1	Jan	Thursday	11	18.5	
4	2	1	Jan	Thursday	11	20.75	
5	2	1	Jan	Thursday	11	16	
6	2	1	Jan	Thursday	11	20.75	
7	3	1	Jan	Thursday	12	16.5	



```
%sql
select
count(distinct order_id) order_id,
sum(quantity) quantity,
date_format(order_date,'MMM') month_name,
date_format(order_date,'EEEE') day_name,
hour(order_time) order_time,
sum(unit_price) unit_price,
sum(total_price) total_sales,
pizza_size,
pizza_category,
pizza_name
from pizza_sales_analysis
group by 3,4,5,8,9,10
```

(6) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [order_id: long, quantity: long ... 8 more fields]

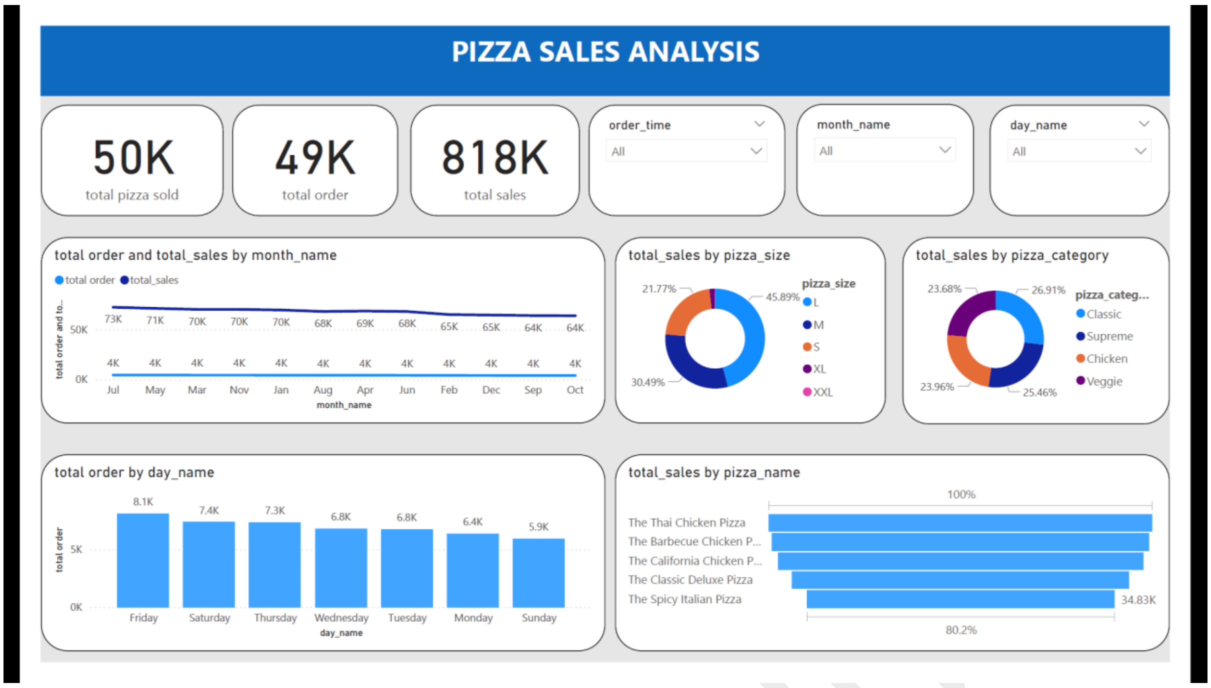
	order_id	quantity	month_name	day_name	order_time	unit_price	total_sal
1	1	1	Feb	Sunday	16	13.25	
2	1	1	Jun	Wednesday	18	16.75	
3	3	3	May	Wednesday	12	53.85000228881836	53.8500022
4	3	3	Feb	Friday	16	62.25	
5	6	6	Jan	Friday	19	72	
6	1	1	Feb	Tuesday	17	17.9500007629394...	17.9500007€

5. **Save the Transformed Data:**
   - Write the processed data back to Blob Storage for Power BI consumption.

Step 5: Visualize Data in Power BI

Objective: Create interactive dashboards to present insights.

1. **Connect Power BI to Azure Databricks:**
   - Install the Databricks ODBC driver.
   - Use the driver to connect Power BI to Databricks.
2. **Build Dashboards:**
   - Create visuals such as:
     - Bar charts for total sales by pizza type.
     - Pie charts showing quantity sold distribution.
     - Line graphs for sales trends over time.
   - Use slicers for filtering data by date or product.
3. **Publish Reports:**
   - Publish the Power BI report to the Power BI service for collaboration.



4. Outcome

Project Deliverables

1. **SQL Server to Azure Data Migration:** Successfully moved on-prem data to Blob Storage using Azure Data Factory.
2. **Data Transformation in Azure Databricks:** Processed raw data into meaningful insights.
3. **Power BI Dashboards:** Developed interactive dashboards showcasing KPIs like total sales and quantity sold.

Key Benefits

- **Automation:** Eliminated manual intervention by setting up a scheduled pipeline.
- **Scalability:** Leveraged Azure's robust infrastructure for future growth.
- **Actionable Insights:** Delivered real-time business metrics for informed decision-making.

5. Conclusion

This project showcases a robust end-to-end Azure data engineering pipeline. By combining the capabilities of SQL Server, Azure Data Factory, Azure Databricks, and Power BI, it provides a scalable solution for data migration, processing, and visualization. These skills are essential for modern data engineers tackling real-world challenges.